#### Angular脱坑记

##### 之同步调用

问题描述

Angular 6.1.5

项目需求：

同步调用因为体验问题，在响应式程序设计中很少采用。但是，在一些特殊的情况下，同步调用方式却可以解决棘手的需求。

比如，一个页面中需要顺序调用若干接口。每个接口的出参，作为下一个接口的入参。最佳解法为采用同步调用方式

解决

resource.ts

public postGetExtCategoryList(): Promise<Object> {  
 return this.postSimpaleAsync(this.getExtCategoryList, {});  
}

返回参数：Promise对象（异步请求返回参数为Observable）

component.ts

async caller() {  
 await this.resourceService.postGetExtCategoryList()  
 .then(res=>{  
 ...

})  
 .catch(()=>{  
 ...

});  
}

await关键字：加异步锁

async关键字：声明异步函数

也可以采用toPromise()方法实现异步调用转同步调用

async initPmModelData() {  
 await this.pmModelResourceService.getResourceTree().toPromise().then(res => {  
 ...

});  
}